

(1) Download VASP

(2) Unpack VASP

(3) Check your Hardware

- (a) A minimum of 256 MB
- (b) Athlon XP, P4 or better
- (c) HD 250 MB free

(4) Check your Compiler and Libraries

- (a) FORTRAN 90 compiler
- (b) BLAS, LAPACK
- (c) Parallel: MPI software

(5) Compile and install VASP

(6) Test VASP



(1) Download VASP

- Make local folder VASP, and subfolder src
- Download VASP:
ftp: //cms.mpi.univie.ac.at
Account: vasp
Password: jk35QGB7
- Potential files for the projector-augmented wave (PAW) method, are located in a separate account on the same ftp server:
ftp: //cms.mpi.univie.ac.at
Account: paw
Password: VgAE4y

Outline of Structure

- VASP/src/vasp.4.lib (Source code; rarely changes)
- VASP/src/vasp.4.6 (Main Fortran 90 code)
- VASP/pot/ ((ultrasoft) pseudopotentials – LDA version)
- VASP/pot_GGA/ ((ultrasoft) pseudopotentials – GGA version)
- VASP/otpaw/ (projector augmented wave potentials - LDA version)
- VASP/otpaw_GGA/ (projector augmented wave potentials – GGA version)

- In VASP: Directories we are interested in : doc, src, pot, pot_GGA
 - /src : Download (a) vasp.4.6.tar.gz
(b) vasp.4.lib.tar.gz
(c) benchmark.tar.gz
(d) benchmark.Hg.tar.gz
 - /pot: Download (a) potcar.Apr00.tar
(b) README.Apr00
 - /pot_GGA : Download (a) potcar.Apr00.tar
(b) README.Apr00
- In paw: Directories we are interested in : potpaw, potpaw_GGA
 - /potpaw: Download (a) potcar.06Feb03.tar
(b) README.UPDATES
 - /potpaw_GGA : Download (a) potcar.06Feb03.tar
(b) README.UPDATES

(2) Unpack VASP

- Unzip in separate folders

In /src:

```
gunzip vasp.4.6.tar.gz (or uncompress vasp.4.6.tar.Z)
```

```
tar -xvf vasp.4.6.tar
```

```
gunzip vasp.4.lib.tar.gz (or uncompress vasp.4.6.tar.Z)
```

```
tar -xvf vasp.4.lib.tar
```

Two directories are created:

vasp.4.lib/ and vasp.4.6/

- Similarly, **mkdir** pot and pot_GGA and unzip files there.

(3),(4) Check hardware, compilers, libraries, etc.

- **BLAS** (Basic Linear Algebra Subprograms) must be installed on the machine. Options:
 - (1) Get the lapackage from <ftp.netlib.org> and compile the blas routines (BLAS/SRC directory). Use g77 or f77 for the compilation.
 - (2) Get an optimized BLAS: For a list of optimized BLAS Try http://www.kachinatech.com/~hjjou/scilib/opt_blas.html

The the two most reliable packages around are presently:

- (a) Intels optimised BLAS (PIII, P4, Itanium)

<http://developer.intel.com/software/products/mkl/>

- (b) Atlas based BLAS routines (for Athlon; mkl routines not optimal)

<http://math-atlas.sourceforge.net/>

(5) Compile and Install VASP

- Go to library directory
`cd vasp.4.lib`
- Look at makefiles (makefile.machine)
e.g. makefile.linux_pg or makefile.linux_ifc_P4
- Choose a makefile and copy to file Makefile
`cp makefile.linux_ifc_P4 Makefile`
- Read Makefile; follow guidelines
- Finally, make the VASP library
`make`

Similarly.....

- Go to vasp.4.6 directory
`cd vasp.4.6`
- Look at makefiles (makefile.machine)
e.g. makefile.linux_pg or makefile.linux_ifc_P4
- Choose a makefile and copy to file Makefile
`cp makefile.linux_ifc_P4 Makefile`
- Read Makefile; follow guidelines
- Finally, make VASP
`Make`
- IMPORTANT!!!! Check if VASP was created
`ls -l vasp`

(6) Test VASP

- Create a new directory bench and untar benchmark.tar.gz
- Open INCAR file

vi INCAR

- Change the file line with IALGO = 8 to IALGO = 48
- Start VASP

vasp.4.6/vasp

```
....  
< DAV: 4 -0.902263882471E+04 -0.11847E-04 -0.88141E-06 56 0.459E-02  
< 5 T= 1918. E= -.90209007E+04 F= -.90226388E+04 E0= -.90218783E+04 EK=  
0.17354E+01  
SP= 0.26E-02 SK= 0.94E-04  
---  
> CG : 1 -0.902264199383E+04 -0.90226E+04 -0.25955E-03 99 0.166E-01 0.241E-02  
> CG : 2 -0.902264200818E+04 -0.14350E-04 -0.45941E-05 49 0.354E-02  
> 5 T= 1918. E= -.90209037E+04 F= -.90226420E+04 E0= -.90218816E+04 EK=  
0.17356E+01  
SP= 0.26E-02 SK= 0.94E-04  
.....
```

Full output can be found in file OSZICAR.ref_4.6

Files Used in VASP

- INCAR input **IMPORTANT**
- STOPCAR input
- stout output
- POTCAR input **IMPORTANT**
- KPOINTS input **IMPORTANT**
- IBZKPT out
- POSCAR input **IMPORTANT**
- CONTCAR output
- EXHCAR input
- CHGCAR input /output
- CHG out
- WAVECAR input /output
- TMPCAR input /output
- EIGENVAL output
- DOSCAR output
- PROCAR output
- SZICAR output
- CDAT output
- XDATCAR output
- LOCPOT output
- ELFCAR output
- PROOUT output